

Amendments to the Claims

The listing of claims below will replace all prior versions and listings of claims in the application.

1. (Currently Amended) A method for minimizing an amount of data needed to test a geometry chunk in a frame against subarea boundaries in a compositing window, comprising the steps of:

defining the geometry chunk with a bounding region, wherein said bounding region defines a space the geometry chunk occupies on the compositing window;

storing data that defines said bounding region for use in processing the geometry chunk in a subsequent frame;

sending said data that defines said bounding region to graphics pipelines;

~~determining~~ determining, from said data that defines said bounding region, a graphics pipeline of said graphics pipelines that will render the geometry ~~chunk defined by said bounding region;~~ chunk;

assigning a subarea in the compositing window to receive an output of said graphics pipeline; and

communicating data associated with the geometry chunk to said graphics ~~pipeline that will render the geometry chunk;~~ pipeline;

wherein said graphics pipelines are configured to render the frame by spatial compositing through parallel ~~processing;~~ processing, said data that defines said bounding

region is less than said data associated with the geometry chunk, and the geometry chunk is different from said subarea.

2. (Previously Presented) The method of claim 1, wherein the geometry chunk is comprised of a piece of a geometry provided by a graphics application.
3. (Previously Presented) The method of claim 1, wherein said space is a screen space.
4. (Previously Presented) The method of claim 1, wherein said space is a world space.
5. (Previously Presented) The method of claim 1, wherein said space is an object space.
6. (Canceled)
7. (Canceled)
8. (Original) The method of claim 1, wherein the geometry chunk is represented as a display list.
9. (Original) The method of claim 1, wherein the geometry chunk is represented as a vertex array object.

10. (Original) The method of claim 1, wherein the geometry chunk is represented as buffered vertices.

11. (Currently Amended) A system for minimizing an amount of data needed to test a geometry chunk in a frame against subarea boundaries in a compositing window, comprising:

graphics pipelines;

a geometry distributor ~~that defines~~ configured to define a bounding region for the geometry chunk, ~~sends to send data that defines~~ said bounding region to said graphics pipelines to ~~determine~~ determine, from said data that defines said bounding region, a graphics pipeline of said graphics pipelines that will render the geometry ~~chunk defined by said bounding region~~, ~~assigns a subarea in the compositing window to receive an output of said graphics pipeline, chunk, to assign a subarea in the compositing window to receive an output of said graphics pipeline, and communicates to communicate data associated with the geometry chunk to said graphics pipeline that will render the geometry chunk;~~ pipeline; and

a memory ~~that stores~~ configured to store said data that defines said bounding region for use in processing the geometry chunk in a subsequent frame;

wherein said bounding region defines a space the geometry chunk occupies on the compositing window;

wherein said graphics pipelines are configured to render the frame by spatial compositing through parallel ~~processing;~~ processing;

wherein said data that defines said bounding region is less than said data associated with the geometry chunk; and

wherein the geometry chunk is different from said subarea.

12. (Currently Amended) The system of claim 11, further comprising a graphics application ~~that provides~~ configured to provide the geometry chunk to said geometry distributor.

13. (Currently Amended) The system of claim 12, wherein said geometry distributor comprises a virtual graphics unit ~~that interfaces~~ configured to interface with said graphics application.

14. (Previously Presented) The system of claim 11, wherein the geometry chunk is comprised of a piece of a geometry provided by a graphics application.

15. (Previously Presented) The system of claim 11, wherein said space is a screen space.

16. (Previously Presented) The system of claim 11, wherein said space is a world space.

17. (Previously Presented) The system of claim 11, wherein said space is an object space.

18. (Currently Amended) The system of claim 11, wherein said geometry distributor comprises:

a bounding region calculator ~~that calculates~~ configured to calculate said bounding region for the geometry chunk;

a graphics pipeline assignor ~~that assigns~~ configured to assign said graphics pipeline to said subarea in the compositing window; and

a graphics pipeline distributor ~~that distributes~~ configured to distribute the geometry chunk to the graphics pipeline.

19. (Original) The system of claim 11, wherein the geometry chunk is represented as a display list.

20. (Original) The system of claim 11, wherein the geometry chunk is represented as a vertex array object.

21. (Original) The system of claim 11, wherein the geometry chunk is represented as buffered vertices.